



# **WASTE - PALAUTETTUJEN TUOTTEIDEN SEURANTAJÄRJESTELMÄ**

Anssi Kulju

Opinnäytetyö  
Toukokuu 2011  
Tietotekniikan koulutusohjelma  
Ohjelmistotekniikka  
Tampereen ammattikorkeakoulu

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietotekniikan koulutusohjelma  
Ohjelmistotekniikka

KULJU, ANSSI: WASTE - palautettujen tuotteiden seurantajärjestelmä

Opinnäytetyö 31 s., liitteet 4 s.  
Toukokuu 2011

---

Tässä työssä kehitettiin vaatimusmäärittelyn pohjalta järjestelmä, jolla voidaan ohjata ja seurata palautetuille tuotteille tehtyjä toimenpiteitä. Järjestelmä rakennettiin Veljekset Keskinen Oy:n käyttöön. Tavoitteena oli yhtenäistää erilaiset käytännöt takuuprosessien hoitamisessa sekä vähentää reklamaatioista aiheutuvaa työtä. Järjestelmä suunniteltiin käyttäjien toiveiden mukaisesti tarkoituksena tehdä mahdollisimman monipuolinen ja helppokäyttöinen työkalu tuotepalautusprosessien seuraamiseen ja hallintaan. Tärkeimpinä suunnittelulähtökohtina olivat virhetilanteiden ennakointi ja niiden välttäminen sekä tietojen pitäminen kaikkien saatavilla, jotta tuotteen käsittely ei pysähtyisi vastuussa olevan henkilön poissa ollessa. Järjestelmän nimi WASTE on akronyymi sanoista Warranty And Service Table Entries.

WASTE pyrittiin ottamaan testikäyttöön niin nopeasti kuin mahdollista, jotta mahdolliset viat ja lisäkehitystä vaativat ominaisuudet tulisivat heti ilmi. Ensimmäinen käyttöönotetussa versiossa oli vain välttämättömimmät toiminnot, kuten tietojen kirjaaminen ja takuukäsittelyiden tilaseuranta mutta jo nämä ominaisuudet riittivät monen tuotteen koko tuotepalautusprosessin hoitoon. Kun puutteet saatiin korjattua, voitiin järjestelmä ottaa laajemmin tuotantokäyttöön.

Ilman hyvin organisoitua ja keskitettyä järjestelmää reklamaatietietojen käsittely ja arkistointi on vaikeaa ja työlästä. Työtä lisää myös se, että usein samoja tietoja joudutaan kirjaamaan useampaan kertaan, koska niitä ei ole missään valmiina. WASTE:ssa suurin työ on käyttöönoton yhteydessä tehtävä alkutietojen kirjaaminen järjestelmään. Kun tämä on tehty, ohjelma tietää, miten mikäkin tuote käsitellään ja kysyy palautuskäsittelyn alussa ainoastaan tarpeelliset tiedot prosessista. Käyttäjän ei esimerkiksi tarvitse itse tietää, mitä tuotteelle tapahtuu vastaanottamisen jälkeen tai mihin huoltoliikkeeseen tuote pitäisi lähettää.

---

Asiasanat: Takuu, hyvitys, reklamaatio, symfony, PHP.

## ABSTRACT

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Programme in information technology  
Option of Software Engineering  
KULJU, ANSSI: WASTE - Tracking System For Returned Products  
Bachelor's thesis 31 pages, appendices 4 pages  
May 2011

---

The purpose of this thesis was to create a centralized system for product reclaim control. It was developed for Veljekset Keskinen Oy's use. The objective was to increase effectiveness in product reclaim treatment including service and refund tracking. The project consisted of designing and implementing the system based on the given specification. It was named WASTE which is an acronym for Warranty And Service Table Entries.

WASTE was deployed to test use shortly after the first critical features were implemented. This was an easy way to recognize the lacks and get ideas from end users. In some cases the first version was already versatile enough to carry out the entire reclaim process from the beginning to the end. When the most important missing features had been added the system was ready to be introduced more widely in the company.

WASTE was designed to make handling of product reclaims as easy as possible and reduce the amount of work it needs. This was accomplished by minimizing data redundancy and implementing the product control so that user does not have to be aware of the way how the product should be dealt with. WASTE gives all needed instructions to user based on maintenance info that must be entered in the system beforehand.

---

Key words: Warranty, refund, reclaim, symfony, PHP.

# SISÄLLYS

|   |    |
|---|----|
| 1 JOHDANTO .....                                      | 6  |
| 2 VAATIMUKSET.....                                    | 7  |
| 3 JÄRJESTELMÄN TEKNIikka.....                         | 8  |
| 3.2 Symphony .....                                    | 8  |
| 3.2.1 Symphony-sovelluksen toiminta .....             | 9  |
| 3.2.2 Doctrine .....                                  | 10 |
| 4 JÄRJESTELMÄN SUUNNITTELU .....                      | 12 |
| 4.1 Tietokantarakenne.....                            | 12 |
| 4.2 Käyttäjätilit .....                               | 13 |
| 4.3 Vastualueet.....                                  | 14 |
| 4.4 Käsittelyn tilat.....                             | 15 |
| 4.5 Tapahtumat .....                                  | 16 |
| 4.6 Integrointi.....                                  | 16 |
| 4.7 Käyttötapaukset .....                             | 17 |
| 4.8 Käyttöliittymä .....                              | 18 |
| 4.9 Lisäominaisuudet .....                            | 19 |
| 4.9.1 RMA-hallinta .....                              | 19 |
| 4.9.2 Hyvitysseuranta.....                            | 19 |
| 5 JÄRJESTELMÄN TOTEUTUS .....                         | 21 |
| 5.1 Suunnittelutulosten sovittaminen käytäntöön ..... | 21 |
| 5.2 Toteutuksessa kohdatut haasteet .....             | 22 |
| 6 KÄYTTÖÖNOTTO.....                                   | 23 |
| 6.1 Testaaminen .....                                 | 23 |
| 6.2 Käyttöönotto .....                                | 23 |
| 6.3 Käyttäjien toiminta jatkossa.....                 | 24 |

|                  |    |
|------------------|----|
| 7 POHDINTA ..... | 26 |
| LÄHTEET.....     | 27 |
| LIITTEET.....    | 28 |

## 1 JOHDANTO

Suuren asiakasmäärän mukana tulee myös paljon reklamaatioita. Tällöin on tärkeää, että palautettuja tuotteita pystyy seuraamaan ja että ne saadaan mahdollisimman nopeasti huoltoon ja takaisin asiakkaalle. Jokaisen myyjän olisi tarvittaessa pystyttävä näkemään, missä palautettu tuote on, mitä sille on tehty ja kuka asiaa on hoitanut. Jos asiakkaalle pystytään välittömästi kertomaan tuotteen tila eikä tietojen selvittämiseen tarvita käsittelystä vastuussa olevan henkilön läsnäoloa, asiakaspalvelusta tulee mahdollisimman sujuvaa. Tähän asti takuuasiat on hoidettu enimmäkseen kirjaamalla tarvittavat tiedot paperille tai paikalliselle tietokoneelle tekstitiedostoon. Tiedot voivat olla hyvin epämääräisiä, ja jos reklamaatioita hoitanut henkilö on lomalla tai poissa, käsittelyt saattavat pitkittyä. Lisäksi syntyy hävikkiä, jos uuteen vaihdettuja tuotteita jää hyllyihin eikä niitä toimiteta asianmukaisesti hyvitetäviksi. Myös huoltoliikkeiden ja toimittajien yhteystiedot pitäisi aina olla helposti saatavilla, jotta voitaisiin esimerkiksi ohjata asiakasta ottamaan yhteyttä suoraan takuunantajaan.

Tämän työn tavoitteena on yhtenäistää erilaiset takuu- ja palautuskäytännöt sekä selkeyttää reklamaatiosta aiheutuvaa prosessia. Työn tarkoituksena on suunnitella ja toteuttaa tähän tarkoitukseen sopiva järjestelmä annetun vaatimusmäärittelyn perusteella.

Tässä dokumentissa on kerrottu, mitä toiminnallisuutta järjestelmältä vaadittiin sekä miten järjestelmä suunniteltiin ja toteutettiin. Vaatimusmäärittely on tehty keräämällä takuuasioita hoitavien henkilöiden kokemuksia nykyisestä takuukäytännöstä. Dokumentti ei esitellä ohjelmakoodia, vaan kuvataan järjestelmää suunnittelun ja ongelmanratkaisun kannalta. Työssä käytetty erityissanasto on selitetty liitteessä 1.

## 2 VAATIMUKSET

Järjestelmän tulisi yhtenäistää tuotteiden palautuskäytäntöä siten, että kaikki takuukäsittelyn aikana tarvittavat tiedot olisivat helposti saatavilla yhdessä paikassa ja jokaisella myyjällä olisi mahdollisuus nähdä myös toisten kirjaamat tiedot. Ohjelmiston tulisi nopeuttaa ja selkeyttää takuuprosessia, jotta palautetut tuotteet saataisiin mahdollisimman nopeasti asianmukaiseen huoltoon ja takaisin asiakkaalle. Alkuperäiset toiminnalliset vaatimukset on esitetty seuraavassa luettelossa. Yksi käsittely vastaa yhden tuotteen palautuksesta aiheutuvaa prosessia.

- Käsittelyllä tulee olla aina yksi tai useampi vastuhenkilö, joka vastaa siitä, että käsittely etenee aikataulun mukaan.
- Käyttäjät yksilöidään siten, että jokainen järjestelmään syötetty tieto on jäljitettävissä.
- Jos käsittely viivästyy, vastuhenkilölle lähtee automaattinen sähköposti-ilmoitus.
- Tuotteita on pystyttävä hakemaan useilla eri tiedoilla niin, että tulokset näkyvät jo kirjoitusvaiheessa.
- Järjestelmästä on pystyttävä hakemaan huoltoliikkeiden yhteystietoja.
- Ohjelman avulla on pystyttävä tulostamaan huoltoon tarvittava lähete sekä asiakkaalle annettava kuitti, jossa on listattuna huollossa tehdyt toimenpiteet, mahdollinen hinta, sekä viivakoodi kassaa varten.

Ei-toiminnalliset vaatimukset liittyvät järjestelmän tekniikkaan. Ohjelman on oltava web-pohjainen, ja sitä on pystyttävä käyttämään yrityksen käytössä olevalla Internet Explorer 8 -selaimella. Lisäksi vanhoja käsittelytietoja on pystyttävä tarpeen vaatiessa hakemaan jälkikäteen.

### 3 JÄRJESTELMÄN TEKNIikka

#### 3.1 Sovellusalusta

Järjestelmää ajetaan LAMP-palvelimella sen joustavuuden ja helpon ylläpidon vuoksi. Ohjelma toteutetaan pääosin PHP-kielellä ja se rakennetaan Symfony-sovelluskehikseen. Sovelluskehiksen käyttämisessä etuna on se, että yleisimmät web-sovelluksissa tarvittavat perusasiat, kuten lomakkeiden virhekäsittely ja tietokanta-abstraktio on usein toteutettu ja testattu valmiiksi, joten suurin osa ajasta voidaan keskittää järjestelmän oman toiminnallisuuden suunnitteluun. Hakuihin käytetään Zend Frameworkiin kuuluvaa ja helposti integroituvaa Lucene-hakukonetta, joka indeksoi tietokannan tiedot hakujen nopeuttamiseksi. Lisäksi haku tehdään inkrementaaliseksi AJAXilla, jolloin on mahdollista siirtää tietoa asynkronisesti palvelimen ja selaimen välillä ilman sivun uudelleenpäivittämistä.

#### 3.2 Symfony

Symfony on yrityskäyttöön suunnattu ilmainen ja avoimen lähdekoodin web-sovelluskehys PHP-projekteille. Se sai alkunsa Mojavi 3:n kehityshaarana, josta se kehittyi itsenäiseksi projektiksi. Ensimmäisen version julkaisi Sensio Labsin perustaja Fabien Potencier lokakuussa 2005. Hänen alkuperäinen tarkoituksensa oli kehittää Symfony lähinnä yrityksensä käyttöön. Kuitenkin muutaman onnistuneen projektin jälkeen hän päätti julkaista sen avoimen lähdekoodin MIT-lisenssillä. Potencier toimii yhä projektin pääkehittäjänä. (Zaninotto, F & Potencier 2010, 12 - 13.)

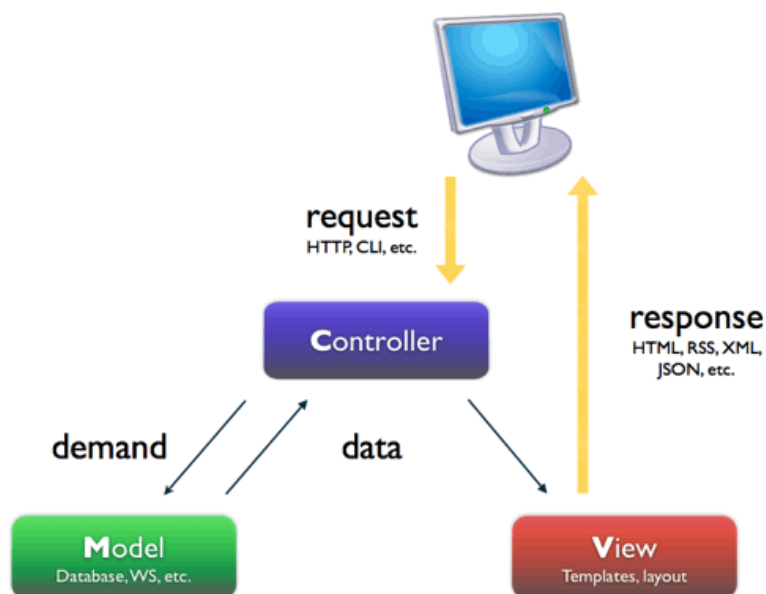
Symfonyn rakenne perustuu web-ohjelmoinnissa paljon käytettyyn MVC-malliin, jossa näkymät erotetaan muusta logiikasta. Se tarjoaa laajan valikoiman erilaisia ominaisuuksia ja työkaluja, jotka nopeuttavat ja selkeyttävät web-sovellusten kehittämistä. Tällaisia ovat 2 ORM-rajapintaa, lomakkeiden validointi ja XSS- ja CSRF-suojaukset, sisäänrakennettu testauskehys sekä monipuoliset konfigurointi- ja laajennusmahdollisuudet. Symfonyn komponentit ovat hyvin vähän riippuvaisia toisistaan, joten suurin



osa niistä soveltuu käytettäväksi erillään sovelluskehiksestä. Lisäksi se on hyvin dokumentoitu ja tukee useita eri tietokantamoottoreita. Symfony on kirjoitettu kokonaan PHP:llä. (Zaninotto, F & Potencier 2010, 11 - 12, 42.)

### 3.2.1 Symfony-sovelluksen toiminta

Sovellus jaetaan tyypillisesti useampaan osaan eli moduuliin. Jokaisella moduulilla on MVC-rakenteen mukaisesti malli (model), näkymä (view) ja ohjain (controller) kuvion 1 mukaisesti. Kun käyttäjä avaa minkä tahansa sovelluksen sivuista, web-pyyntö päätyy aina sovelluskohtaiselle etuohjaimelle, joka välittää pyynnön eteenpäin suodatin-ketjun läpi asianmukaisen moduulin toimintametodille (action). Moduulin oletustoiminnot koostuvat objektin luonnista, poistamisesta, muokkaamisesta, tietojen näyttämisestä ja objektilistauksesta. Toimintoja ei kuitenkaan suoriteta ohjainkoodissa, vaan tehtävä delegoidaan malliluokille, joissa voidaan käsitellä tietokannassa olevaa tietoa.



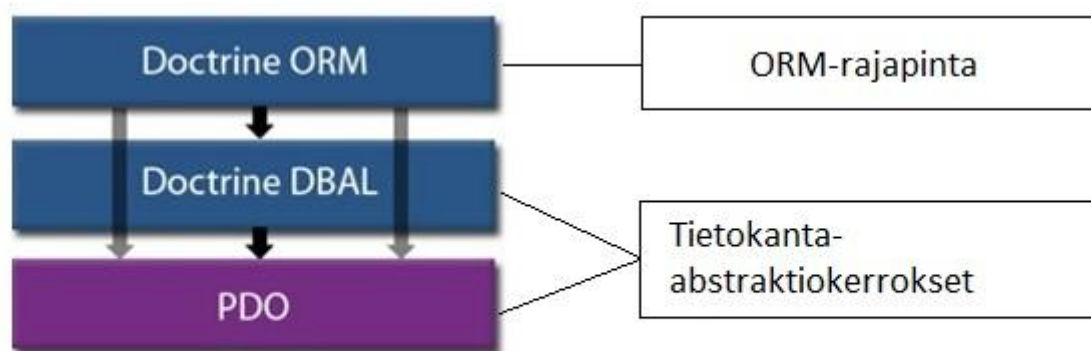
KUVIO 1: Symfony:n MVC-arkkitehtuuri (Potencier 2009, 44)

Mallissa haetut ja prosessoidut tiedot sovitetaan toimintametodissa määrättyyn näkömähjaan ja valmis sivu lähetetään käyttäjälle. Jokaisella toiminnolla voi olla omat näkymänsä, joiden sisältö koostuu enimmäkseen selaimen ymmärtämästä HTML-datasta. PHP:tä käytetään näkymän koodissa yleensä vain tietojen tulostamisessa tarvittaviin toimintoihin, kuten ehtoihin ja toistoihin. Näkymään voi liittää tarpeen mukaan

CSS-tyyliohjeita ja selaimessa käytettävää Javascript-koodia toiminnallisuuden lisäämiseksi. Ennen käyttäjälle lähettämistä, näkymä liitetään osaksi sovelluskohtaista näkymäpohjaa, joka voi sisältää esimerkiksi otsikon tai logon.

### 3.2.2 Doctrine

Doctrine on projektissa käytettävä ORM-tekniikka. Sen tehtävänä on tarjota relaatiotietokannalle objektimainen ohjelmointirajapinta, jolloin koodi soveltuu paremmin olio-ohjelmointiin. Doctrine ORM on toteutettu kahden abstraktiokerroksen päälle kuvion 2 osoittamalla tavalla, joten tietokantajärjestelmän SQL-kieltä ei tarvitse käyttää suoraan, vaan kyselyt voidaan toteuttaa Doctrinen omalla DQL-kyselykielellä (Wage, Borschel & Blanco 2010, 177). Kun tietokannasta haetut tiedot on tallennettu PHP-objekteihin, voidaan tietoja muokata ja tallentaa sopivien metodien avulla. Relaatiotietokannassa yksi taulurivi vastaa yhtä Doctrinen tietueobjektia. Tiedon haku- ja kirjoitusmetodeihin voidaan helposti sisällyttää normaalia enemmän toiminnallisuutta ja niitä voi tarvittaessa tehdä itse. Oliomaisuuden ansiosta voidaan metodeille toteuttaa myös yksikkötestejä.



KUVIO 2: Doctrinen abstraktiokerrosrakenne (Wage, Borschel & Blanco 2010, 29, muokattu)

Tietokannan taulut luodaan YAML-muotoisen skeematiedoston avulla. Doctrine generoi tarvittavat lauseet skeematiedoston mukaisesti optimoituina käytössä olevalle tietokantajärjestelmälle ja luo samalla tauluja vastaavat PHP-luokkapohjat mallia varten. Näin suorituskykyä saadaan nostettua ja tietokantaa on helpompi ylläpitää järjestelmän kehitysvaiheessa, koska mahdolliset rakennemuutokset tehdään ainoastaan skeematiedostoon. YAML-muotoisia tiedostoja käytetään myös testidatan syöttämisessä tietokantaan. (Potencier, F. 2009, 36 - 38.)

## 4 JÄRJESTELMÄN SUUNNITTELU

Järjestelmän suunnittelussa kantavana ajatuksena on takuuasioiden hoidosta aiheutuvan työmäärän vähentäminen ja helpottaminen. Jos työmäärä lisääntyy ohjelman käyttöönoton takia, järjestelmää ei pidetä hyödyllisenä ja vanhat takuukäsittelymenetelmät mielletään paremmiksi. Järjestelmää käytetään myös asiakaspalvelutilanteissa, joten tuotteen vastaanottamisen tulee olla mahdollisimman sujuva prosessi. Tätä edesauttaa kaiken mahdollisen aikaa vievän työn siirtäminen ennakkoon tehtäviin ylläpitotoimintoihin. Työtä pystytään helpottamaan myös selkeän käyttöliittymän ja mahdollisimman vaivattoman hakutoiminnon avulla. Toteutuksen kannalta järjestelmä suunnitellaan siten, että sitä on helppo laajentaa, ylläpitää ja testata. Tällä tavalla uudet ominaisuudet saadaan mahdollisimman nopeasti ideasta toteutukseen ja käyttöön.

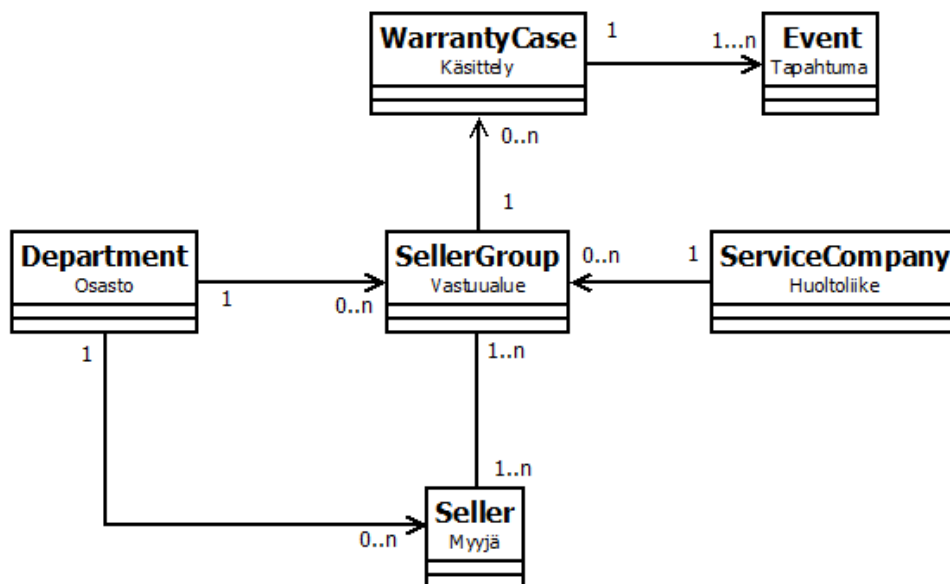
### 4.1 Tietokantarakenne

Järjestelmän perimmäinen tarkoitus on tiedon säilyttäminen helposti haettavassa ja luettavassa muodossa. Suunnittelun tärkeimpänä vaiheena voidaan siis pitää tietokantarakenteen mallintamista. Tietokantarakenne on esitetty kuviossa 3 luokkakaavion avulla. Seuraavassa kappaleessa on esitetty, millä perustein tähän luokkarakenteeseen päädyttiin.

Kuten vaatimuksissa on esitetty, käyttäjätiedot pitää yksilöidä siten, että jokainen syötetty tieto on jäljitettävissä, joten tarvitaan yksittäisen käyttäjän tietoja kuvaava luokka. Lisäksi luodaan osaston tietoja varten oma luokkansa, jota voidaan käyttää sekä käyttäjien että käsittelyiden ryhmittelyyn. Asiakastietoja ei tarvitse pitää yllä, joten ne voidaan tallentaa jokaisessa käsittelyssä erikseen. Tämän vuoksi asiakastiedot ja käsittelytiedot voidaan sijoittaa samaan käsittelyä kuvaavaan luokkaan. Käsittelyiden tilaa pitää pystyä seuraamaan, joten on myös tärkeää tietää, milloin tila on vaihtunut ja kuka on vaihdoksen tehnyt. Tähän tarkoitukseen kehitetään tapahtumat, joita luodaan aina kun käsittelyihin tulee muutoksia. Käsittelyt jaotellaan osastojaon lisäksi vastuualueisiin, joihin on määritelty vastuuhenkilöt. Vastuualueilla on myös tietty huoltoliike tai toimit-

taja, johon ollaan yhteydessä alueeseen kuuluvien tuotteiden huolto- ja hyvitysasioita selviteltäessä. Vastuualueita ja huoltoliiketietoja varten luodaan myös vastaavat luokat.

Kuvion 3 luokkakaaviosta nähdään myös luokkien suhteet toisiinsa. Kaaviosta nähdään, että käsittelyllä tulee olla aina vähintään 1 tapahtuma. Tämä tapahtuma luodaan heti käsittelyn luonnin yhteydessä. Lisäksi myyjälle ja vastuualueelle tulee aina olla määritelty osasto, mutta myyjä voi olla myös muun osaston vastuualueen vastuuhenkilö.



KUVIO 3: Tietokannan rakennetta kuvaava luokkakaavio

## 4.2 Käyttäjätilit

Seller-luokka kuvaa järjestelmän käyttäjää. Käyttäjärekisteriin on tallennettu käyttäjien yhteystiedot ja käyttöön vaadittavat tunnukset ja salasanojen SHA1-tiivisteet. Ohjelma ei käyttäjätunnuksista huolimatta vaadi sisäänkirjautumista, koska tämä hankaloittaisi työtä asiakaspalvelutiskillä myyjän usein vaihtuessa. Sen sijaan käyttäjätunnus ja salasana kysytään tietojen tallentamisen yhteydessä aina erikseen. Salasanalla voi vaihtaa jälkikäteen käyttäjähallinnan kautta. Salasanan unohtuessa käyttäjällä on mahdollisuus pyytää järjestelmästä uusi salasana sähköpostiin.

### 4.3 Vastuualueet

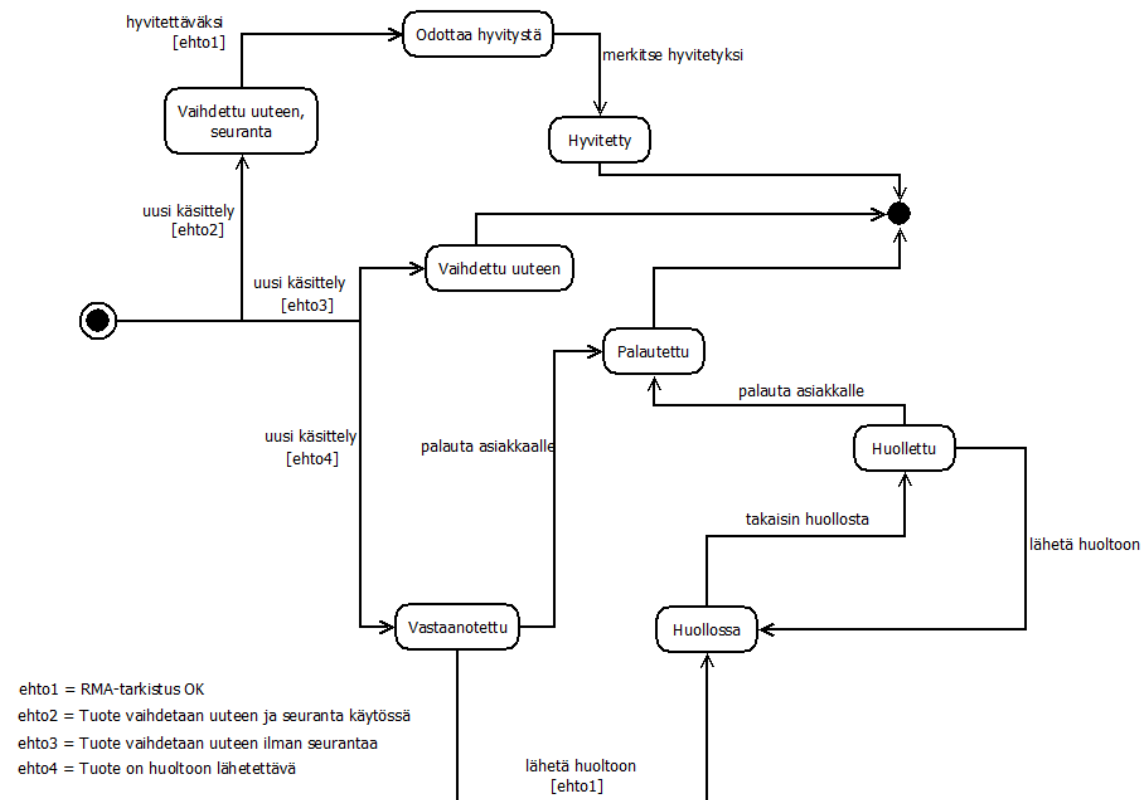
Määrittelyn mukaan jokaisella käsittelyllä tulee olla vähintään yksi vastuuhenkilö. Tämä toteutetaan siten, että ne käyttäjät, joille on määrätty vastuu tiettyjen tuotteiden rek-lamaatioista, sijoitetaan vastaaviin vastuualueisiin. Vastuuhenkilöiden lisäksi alueeseen on määritelty huoltoliike, tietoa tuotteiden käsittelytavasta, alueen tuotteita yhdistävä ja kuvaava nimi sekä aika, jonka jälkeen järjestelmä tulee ilmoittaa vastuuhenkilöille käsittelyn viivästymisestä. SellerGroup-luokka kuvaa vastuualueen tietoja.

Suuren tuotemäärän takia vastuualueita ei sidota tuotteisiin etukäteen, vaan käsittelyä aloittaessa käyttäjä valitsee oikean vastuualueen listasta. Koska käyttäjä ei välttämättä tiedä, kuka kyseisen tuotteen takuuasioista vastaa, tulee vastuualueella olla selkeä ja tuotteita hyvin kuvaava nimi. Kun alue on valittu, ohjelma kysyy muut tiedot sen mukaan, mitä vastuualueeseen on määritetty tarpeellisiksi. Jos esimerkiksi alueen tuotteet ovat uuteen vaihdettavia ja niitä kerätään useampi samaan pakettiin ennen toimittajalle lähetystä, lähetelistaan ei välttämättä tarvitse liittää asiakastietoja. Tässä tapauksessa niiden kysyminen tuotteen vastaanottotilanteessa olisi turhaa ja aikaa vievää. Vastuualueen tarkoitus on näin ollen toimia myös lomakepohjana ja ohjeena tietojen syöttämisessä.

Alueen mukaan voi tuotteen huoltaminen vaatia tarkempaa lisätietoa, kuin lomakkeen kentistä selviää. Esimerkiksi tietokoneiden huoltolähetteessä on käytävä ilmi, otetaanko kiintolevyn sisällöstä varmuuskopio ja mikä käyttöjärjestelmä koneessa on. Tämän kaltaiset tarkennukset vaatisivat jokaiselle vastuualueelle erilaiset huoltolähteet, mistä seuraisi vaikea ylläpidettävyys ja käytön monimutkaistuminen. Erilaisten huoltolähet-teiden sijaan käytetään vastuualuekohtaista vapaamuotoista kysymyslistaa, jossa on lueteltu asiakkaalta kysyttävät pakolliset lisätiedot. Kun kysymykset on määritelty vastuualueen tietoihin, näkyvät ne uuden käsittelyn kirjausvaiheessa lisätietokentän vieres-sä. Listassa lueteltujen pakollisten tietojen lisäksi käyttäjä voi lisätä kenttään myös muuta tarpeellista tietoa tuotteen käsittelystä. Käsittelyn vastuualuetta voi myös vaihtaa jälkikäteen.

#### 4.4 Käsittelyn tilat

Käsittelyillä on aina tuotteen sijaintia kuvaava tila. Tällöin ne on helppo järjestellä esimerkiksi avoimiin ja valmiisiin tapauksiin ja käyttäjä voi helposti päätellä, missä tuote on menossa. Kuviossa 4 on tilakaavio, josta nähdään käsittelyn mahdolliset tilat ja siirtymät. Tilakaaviossa ei ole huomioitu käyttäjätunnistautumisen onnistumista, koska se on automaattisesti jokaisen tilasiirtymän ehto.



KUVIO 4: Tilakaavio

Käyttäjän ei välttämättä tarvitse erikseen päättää käsittelyä. Jos esimerkiksi tuote toimitetaan asiakkaalle suoraan huoltoliikkeestä, käsittely päätetään automaattisesti jo huoltoon lähetyksen yhteydessä. Koska tuotteen palautustapa vaihtelee usein, sitä ei määritellä vastuualueessa, vaan se päätetään käsittelyn aloituksessa. Poikkeuksena ovat uuteen vaihdettavat tuotteet, joissa palautustapa määräytyy automaattisesti erikoistapaukseksi eikä sitä voi valita tai muuttaa.

## 4.5 Tapahtumat

Käsittelyn tilan vaihtuessa tai tietojen muuttuessa järjestelmä luo automaattisesti tietyn tyyppisen tapahtuman, joka näkyy käsittelyn tiedoissa. Event-objekti sisältää tapahtumaan liittyvät tiedot, joihin kuuluvat viesti, aikaleima, tapahtumatyyppi ja tapahtuman aiheuttanut henkilö. Tapahtumatyypit vastaavat käsittelyn tilanmuutoksia, joten käsittelyn tapahtumalistasta on helppo etsiä esimerkiksi tuotteen huoltoonlähetyspäivämäärä ja lähettäjä.

Automaattisten tapahtumien lisäksi käyttäjät pystyvät itse luomaan käsittelyille vapaan viestin sisältäviä tapahtumia eli kommentteja. Kommentoimalla on helppo tiedottaa muita käyttäjiä käsittelyn etenemisestä ja siihen liittyvistä muutoksista.

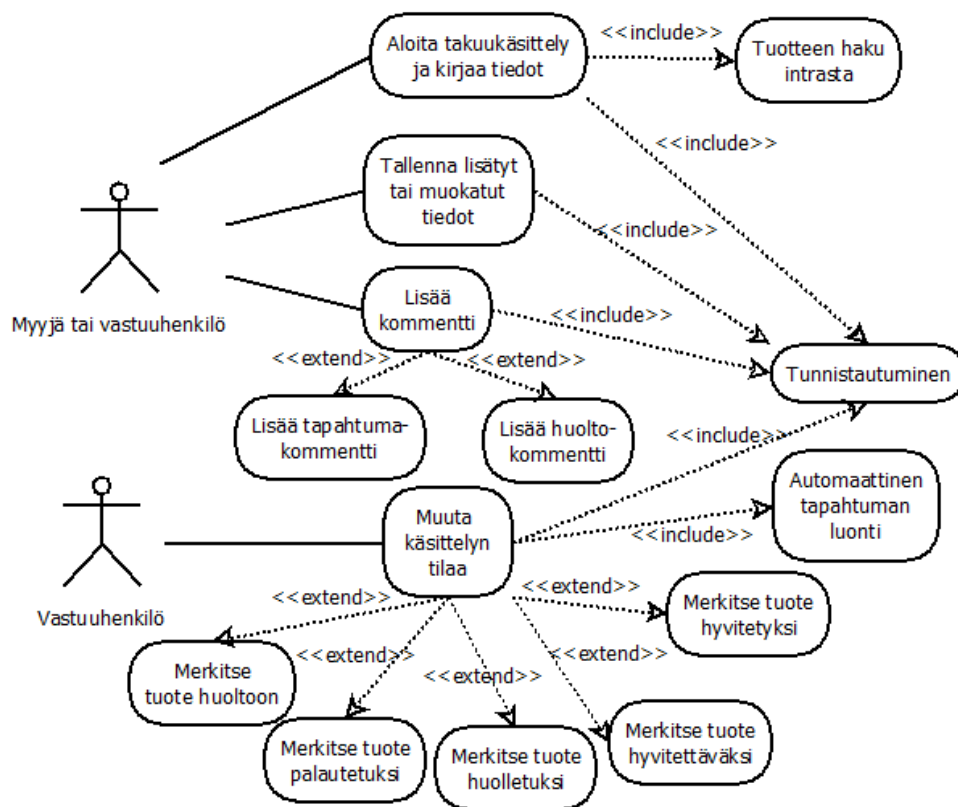
## 4.6 Integrointi

Yrityksellä on käytössä Sipos-kassajärjestelmä, jonka tuotetietokantaa käytetään projektissa hyväksi. Kassaohjelmistoon kuuluu yhtenä osana web-käyttöliittymä Sipos intra, jolla on mahdollista muun muassa käsitellä ja hakea tuotetietoja. Tämän vuoksi takuujärjestelmään ei tehdä uutta tuotehakua vaan ohjelmat integroidaan siten, että uusi takuukäsittely aloitetaan tuotteen hakemisella intrasta ja siirtymällä tuotteen kohdalla olevasta linkistä takuuohjelmaan. Linkin URL sisältää tuotteen EAN-koodin ja kassalinjan tunnuksen, joiden perusteella takuujärjestelmä hakee tuotteen nimen suoraan intran tietokannasta. Käyttäjän näkökulmasta ohjelma toimii siis osana jo ennestään tuttua intraa, mikä puolestaan lisää ohjelman käytön luontevuutta.



#### 4.7 Käyttötapaukset

Järjestelmän kriittisimmät käyttötapaukset on esitetty kaavion avulla kuviossa 5. Kaavioon ei ole sisällytetty käyttöönoton alussa tehtäviä toimenpiteitä, kuten käyttäjätunusten luomista tai huoltoliiketietojen lisäämistä, vaan se keskittyy järjestelmän jokapäiväiseen käyttötapaan.



KUVIO 5: Käyttötapauskaavio käsittelyn hoitamisesta

Käsittelyiden hallinta on jaettu vastualueen hoitajan ja myyjän kesken osiin. Myyjän tehtävänä on ottaa tuotteet vastaan ja lisätä tarvittaessa esimerkiksi asiakkaalta tai huoltoliikkeeltä tulevaa tietoa käsittelyyn. Vastualueen hoitaja huolehtii siitä, että käsittely etenee ajallaan. Tietojen pysyvä poistaminen ei ole mahdollista käyttäjätasolla. Esimerkiksi käyttäjähallinnassa oleva tilin poistamistoiminto ei varsinaisesti tuhoa tilää, vaan piilottaa sen listalta merkitsemällä sen inaktiiviseksi. Myös valmiit takuukäsittelyt säilytetään piilotettuna kannassa. Ne voi asettaa käyttöliittymässä erikseen näkyviin.

## 4.8 Käyttöliittymä

Järjestelmän käyttöliittymä koostuu pääosin lista-, lomake- ja tietonäkymistä. Kuviossa 6 on esitetty järjestelmälle tyypillisen listanäkymän asettelu. Järjestelmällä ei ole tiettyä aloitusnäköä, vaan se riippuu käyttötavasta. Jos luodaan uusi takuukäsittely, ensimmäinen näkymä tuotteen valinnan jälkeen on vastualueen valinta, jota seuraa muiden tarvittavien tietojen syöttäminen. Jos halutaan tarkastella jotain tiettyä tapausta, tai muuttaa tietoja, siirrytään suoraan käsittelylistaan intran etusivulta. Jokaisessa listanäkymässä on hakukenttä, jolla pystytään suodattamaan pois epäoleelliset rivit. Haku toimii reaaliaikaisesti siten, että se näyttää tulokset kolmen ensimmäisen kirjoitetun merkin jälkeen ja karsii tuloksia merkkejä lisättäessä.

| Yläpalkki            |  |                           |  |  |
|----------------------|--|---------------------------|--|--|
| <b>Sivun otsikko</b> |  | Haku <input type="text"/> |  |  |
| Lyhyt selite         |  |                           |  |  |
| Uusi objekti         |  |                           |  |  |
|                      |  |                           |  |  |
|                      |  |                           |  |  |
|                      |  |                           |  |  |
|                      |  |                           |  |  |
|                      |  |                           |  |  |
|                      |  |                           |  |  |
|                      |  |                           |  |  |
|                      |  |                           |  |  |
|                      |  |                           |  |  |
|                      |  |                           |  |  |
|                      |  |                           |  |  |
|                      |  |                           |  |  |
|                      |  |                           |  |  |

KUVIO 6: Yleinen listanäkymä

Listojen ensimmäisessä sarakkeessa on linkit, joista pääsee objektien tietonäkymään. Ylälaidassa olevasta palkista pääsee käsittelylistaan, tapahtumalokiin ja ylläpitovalikkoon. Ylläpitovalikossa on linkit käyttäjä-, vastualue-, osasto- ja yhteystietohallintaan. Käyttöliittymäsuunnittelussa pyritään siihen, että ohjelman käyttö olisi johdonmukaista ja virhetilanteiden mahdollisuus olisi vähäinen. Moduulien eri näkymät pyritään pitämään mahdollisimman samankaltaisina, eli esimerkiksi kaikkiin listanäkymiin on liitetty reaaliaikainen haku ja uuden objektin voi luoda listan yläpuolella olevasta linkistä. Tästä hyödytään myös toteutusvaiheessa, koska koodista voidaan näin tehdä mahdolli-

simman geneeristä ja samaa toimintoa voidaan hyödyntää useassa eri paikassa parantamisen järjestelmän laajennettavuutta ja testattavuutta.

#### 4.9 Lisäominaisuudet

Järjestelmän kehitysvaiheessa tulee ajoittain ideoita ja pyyntöjä tarpeellisista ominaisuuksista. Ideoista keskustellaan käyttäjien kanssa ja tutkitaan, miten lisättävän toiminnallisuuden voisi toteuttaa käytäntöön sopivaksi. Järjestelmän käyttö pyritään kuitenkin edelleen pitämään mahdollisimman yksinkertaisena. Suurimmat tähän mennessä toteutetut lisäominaisuudet ovat RMA-hallinta ja hyvitysseuranta.

##### 4.9.1 RMA-hallinta

RMA:n käyttö määritellään vastuualuekohtaisesti. Jos tämä ominaisuus on aktivoitu, alueeseen kuuluville tuotteille on syötettävä RMA ennen huoltoon lähettämistä. Järjestelmän avulla RMA:n tarvitsevista tuotteista voi kerätä vastuualuekohtaisen listan ja lähettää sen sähköpostitse tavarantoimittajalle RMA-hakemuksen muodossa. Kun RMA on saatu, sen voi syöttää yhtäaikaaisesti listasta valituille tuotteille. Vasta tämän jälkeen kyseiset tuotteet on mahdollista merkitä huoltoon ja jatkaa käsittelyä normaalisti. Listasta nähdään myös, onko tuotteille mahdollisesti jo haettu RMA.

Monilla tavarantoimittajille on oma järjestelmänsä RMA-numeroiden hallintaa varten, jolloin hakemus tehdään yleensä toimittajan omilla web-sivuilla. Vapaamuotoinen sähköpostiviesti ei tällöin välttämättä kelpaa hakemukseksi. Tämän vuoksi asiasta on yleensä sovittava toimittajan kanssa.

##### 4.9.2 Hyvitysseuranta

Joitakin tuotteita ei lähetetä huoltoon, vaan asiakkaalle annetaan takuuvaihtona uusi tuote. Vaihdetut tuotteet palautetaan tai niistä raportoidaan toimittajalle, jolloin niistä on mahdollista saada hyvitys. Hyvityksien seuraamiseksi järjestelmään on kehitetty ominaisuus, joka mahdollistaa tuotteiden seuraamisen uuteen vaihtamisen jälkeen. Seu-

ranta on erikseen aktivoitava vastualueen asetuksista. Käsittelyn tilakierto muistuttaa tällaisissa tapauksissa läheisesti huoltoon lähetettävän tuotteen käsittelyä. Huollon sijaan tuotteet lähetetään hyvitetäväksi, ja kun hyvitys on saatu, merkitään käsittelyjen tilat hyvitettyiksi. Hyvityksiä seurataan vastualueen ylläpidossa, josta nähdään listat uuteen vaihdetuista ja hyvitystä odottavista tuotteista. Listojen avulla käsittelyiden tilamuutokset voidaan kohdistaa koko ryhmälle eikä niitä tarvitse käydä yksitellen läpi.

## 5 JÄRJESTELMÄN TOTEUTUS

### 5.1 Suunnittelutulosten sovittaminen käytäntöön

Sovelluskehityksen käyttö kannustaa jakamaan ja jäsentelemään ohjelmakoodin selkeiksi ja mahdollisimman vähän toisistaan riippuvaisiksi osakokonaisuuksiksi. Tästä on erityisesti hyötyä silloin, kun järjestelmään tehdään muutoksia tai uusia ominaisuuksia. Kun kaikki tiettyyn asiaan liittyvä toiminnallisuus löytyy samasta paikasta, ei tarvitse huolehtia muiden osien rikkoontumisesta. Kehitysvaiheessa radikaaleilta muutoksilta välttyttiin tiiviin yhteistyön ansioista käyttäjien kanssa. Tietokantarakenne pysyi pääosin samanlaisena lukuun ottamatta yksittäisten tietojen lisäyksiä tai poistoja luokkien sisällä. Kuviossa 7 on esitetty WASTE:n pääsivu. Yläreunassa on linkit ylläpitotoimintoihin ja tapahtumiin sekä haku, jolla käsittelyitä voi etsiä listassa näkyvien tietojen perusteella. Oikeassa reunassa olevien nappien avulla voi suorittaa tiettyjä toimintoja suoraan listalta, jolloin ei tarvitse siirtyä erikseen käsittelyn hallintasivulle. Järjestelmän perusrakenne on esitetty kuvan avulla liitteessä 2. Toimintoketju etenee luvussa 3.2.1 esitetyllä tavalla. Kuvassa olevat numerot kuvaavat suoritusjärjestystä.

| WASTE 0.4 (Warranty And Service Table Entries) <a href="#">Uudet ominaisuudet</a> |                 |               |  |                            |                     |  |   |                                       |
|---|-----------------|---------------|--|----------------------------|---------------------|--|---|---------------------------------------|
| <a href="#">Käsittelyt</a>  |                 |               | <a href="#">Tapahtumalogi</a>                  |                            |                     | <a href="#">Ylläpito</a>               |   |                                       |
| <b>Takuukäsittelyt</b>  |                 |               | Listaus ja haku                                |                            |                     | Haku <input type="text"/>              |   |                                       |
| Osasto <input type="text" value="Kaikki"/>  |                 |               | <a href="#">Avoimet</a> <a href="#">Kaikki</a> |                            |                     |  |   |                                       |
| ID  | Asiakas         | EAN           | Tuotteen nimi                                  | Tuotteen tila              | Käsittely aloitettu | Osasto / Vastuualue                    | Pikatoiminnot                                 |                                       |
| 38  | Testi Asiakas 1 | 8715946332758 | EPSON PERFECTION V750 PRO                      | Vastaanotettu              | 16.02.2011 09.58.32 | Autotarvike ja kodinkone / Kodinkoneet | <input type="button" value="Huoltoon"/>       | <input type="button" value="Valmis"/> |
| 37  |                 | 8715946332758 | EPSON PERFECTION V750 PRO                      | Vaihdettu uuteen, seuranta | 15.02.2011 10.23.18 | ATK / testi                            | <input type="button" value="Hyvitettäväksi"/> |                                       |
| 11  | Testi Asiakas 2 | 8712581464431 | PHILIPS 32PFL5604H/12 ANT./KAAP. FULL HD       | Vastaanotettu              | 27.01.2011 08.07.26 | Autotarvike ja kodinkone / Kodinkoneet | <input type="button" value="Huoltoon"/>       | <input type="button" value="Valmis"/> |
| 6   | Testi Asiakas 3 | 6419773639029 | MEISTER LAMINAATTILEIKKURI DAP0031             | Vastaanotettu              | 21.01.2011 11.04.33 | Rauta / Meister                        | <input type="button" value="Huoltoon"/>       | <input type="button" value="Valmis"/> |

KUVIO 7: Valmis käyttöliittymä

## 5.2 Toteutuksessa kohdatut haasteet

Suurimmat ongelmatilanteet järjestelmän toteuttamisessa aiheutuivat Doctrinen muistihallintaominaisuuksista. Doctrine-objektit sisältävät paljon dataa tietokannasta haettujen tietojen lisäksi, jolloin useiden objektien luominen vie huomattavan määrän muistia. Koska järjestelmästä ei ole tarkoitus missään vaiheessa poistaa käsittelytietoja, tulee muistin kulutuksesta ajan myötä ongelma. Tähän mahdollisia ratkaisuja ovat kerralla listattavien käsittelyiden rajaaminen useammalle sivulle. Huolimatta suuri-koisista objekteista, on Doctrinen muistinkäyttö silti säästeliästä. Tämä kävi ilmi käsittelytietojen tallennustilanteessa. Tallennusproseduuri sisältää useita tarkastuksia ja funktiokutsuja, joissa ladataan tietokannasta objekteja PHP:n muistiin. Ennen käsittelyobjektin lopullista tallennusta, tietokantaan luodaan uusi tapahtumaobjekti, jossa puolestaan ladataan sama käsittelyobjekti kannasta uudelleen ilman paikallisia muutoksia. Näin saadaan käsittelyn rivitunnus tapahtuman tietoihin. Kun tapahtuman luonnista palataan takaisin muokatun käsittelyobjektin pariin, myös tämän tiedot ovat palautuneet alkuperäisiksi eikä objekti tallennu uusilla tiedoilla. Vaikka ongelman korjaus vaati kaikkien malliluokkien tutkimisen, oli sen korjaaminen suhteellisen nopea toimenpide verrattuna vian etsimiseen.

## 6 KÄYTTÖÖNOTTO

### 6.1 Testaaminen

Järjestelmä testataan Mozilla Firefoxin Selenium IDE -liitännäisellä, jolla on projektin kehittämisen aikana luotu useita testitapauksia. Symfony sisältää myös oman testikehyksen funktionaaliseen testaamiseen, mutta se osoittautui testien ylläpidon kannalta huomattavasti työläämmäksi Seleniumiin verrattuna. Symfonyyn yksikkötestikehystä, Limeä, käytetään projektissa malliluokkien yksikkötestaamiseen. Limen avulla tietokanta pystytään helposti palauttamaan samaan tilaan testejä varten ja kaikki yksikkötestit voidaan tehdä kerralla automaattisesti.

Funktionaaliset testit toteutetaan harmaalaatikkotesteinä eli siten, että tapaukset suunnitellaan kuvaamaan käyttäjän mahdollista toimintaketjua, jossa kaikki ohjelman kannalta mahdolliset virhetilanteet on tarkastettu. Liitteessä 3 on esimerkki yhdestä testitapauksesta.

### 6.2 Käyttöönotto

Järjestelmän käyttöönottoon kuuluu tuotantopalvelimen valmistelu, tietokannan alustus, ohjelmakoodin siirto sekä asetusten määrittäminen. Tuotantopalvelimella ajetaan PHP-skripti, joka tarkistaa PHP-tulkin käyttämät ominaisuudet ja asetukset, jotta ne soveltuvat Symfony-sovelluksen käyttämiseen. Tuotantoversioon luodaan myös erillinen tietokanta, jotta ohjelmaa pystyisi jatkossakin testaamaan vapaasti kehityksessä käytetyllä tietokannalla.

Projektin siirtämiseen palvelimelle on Symfonyssa oma työkalu, joka muodostaa yhteyden tuotantopalvelimelle ja kopioi sinne rsync:n avulla projektikansion sisällön jättäen pois halutut tiedostot, kuten välimuistitiedostot. Kun järjestelmään on määritelty uudet tietokanta- ja ympäristöasetukset, on se valmiina käytettäväksi.

Kun järjestelmä on toiminnassa, voidaan aloittaa käyttökoulutukset. Osastojen vastuuhenkilöt perehdytetään järjestelmän käyttöön, minkä jälkeen he voivat esitellä ohjelman toimintaa tarpeen mukaan muille työntekijöille. Vastuuhenkilöt luovat käyttäjätunnuksensa järjestelmään ja kirjaavat tarvittavien huoltoliikkeiden yhteystiedot sekä osaston tiedot, jos niitä ei ole valmiina. Tämän jälkeen jokainen vastuuhenkilö voi luoda oman vastuualueensa. Vastuualueen asetukset selvitetään kohta kohdalta ja vastuuhenkilölle kerrotaan, mihin mikäkin valinta vaikuttaa. Lopuksi kokeillaan uuden takuukäsittelyn luontia intran kautta.

### 6.3 Käyttäjien toiminta jatkossa

Järjestelmän käyttötietojen kirjaaminen ja ylläpito ovat osastojen vastuulla. Näihin kuuluvat käyttäjätunnukset, osastotiedot, vastuualueet ja huoltoliikkeiden yhteystiedot. Ennen käyttäjä- ja vastuualuetietojen syöttämistä on luotava vastaava osasto järjestelmään. Tämän jälkeen osaston käyttäjät voivat luoda itse omat käyttäjätunnuksensa. Ne käyttäjät, jotka vastaavat takuuasioista, kirjaavat tarvittavien huoltoliikkeiden yhteystiedot ja suunnittelevat omat vastuualueensa, joihin merkitsevät itsensä vastuuhenkilöiksi. Alustavien toimintojen jälkeen ohjelman avulla voidaan vastaanottaa ja käsitellä palautettuja tuotteita liitteessä 4 olevan aktiviteettikaavion mukaisesti.

Käsittelyn tilaa voi vaihtaa joko tietonäkymässä, tai suoraan listalta pikatoimintojen avulla. Virhetilanteiden vähentämiseksi mahdollisia tilamuutoksia on rajoitettu sen hetkistä tilaa vastaavaksi. Jos tuote on esimerkiksi huollossa, sitä ei voi lähettää uudelleen huoltoon. Valmiiden käsittelyiden tilaa ei pysty jälkikäteen vaihtamaan. Näihin kuuluvat myös ne käsittelyt, joissa asiakas on saanut välittömästi uuden tuotteen eikä hyvitysseuranta ole käytössä. Käsittely merkitään valmiiksi myös huoltoon lähetettäessä silloin, kun tuotteen palautustavaksi on merkitty "Huolto palauttaa tuotteen asiakkaalle".

Kun tuote on merkitty huoltoon, ohjelma näyttää automaattisesti huoltolähetteen, jonka avulla tuote on helppo postittaa eteenpäin. Se sisältää vikaselosteen, yrityksen ja asiakkaan yhteystiedot sekä tiedon siitä, palautetaanko huollettu tuote suoraan asiakkaalle vai takaisin yritykselle. Läheteeseen merkitään lähettäjän yhteystietoihin tilaa vaihta-



neen vastuuhenkilön nimi, puhelinnumero ja sähköpostiosoite sekä yrityksen osoitetiedot. Lähetä on muotoiltu helposti tulostettavaan muotoon ja sen voi tulostaa missä vaiheessa käsittelyä tahansa. Jos tulostusvaiheessa tuote ei ole vielä käynyt huollossa, lähetteen lähettäjäosan tiedoissa huoltoon lähettämisestä vastuussa olevan henkilön tiedot on korvattu osaston yhteystiedoilla.

## 7 POHDINTA

Tuotteiden takuukäsittelytavat riippuvat tavarantoimittajasta mutta usein myös saman toimittajan eri tuotteet, tai joskus jopa tuotemerkin eri mallit hoidetaan eri tavalla. Vastuualueiden suunnittelu on tällaisissa tapauksissa vaikeaa, koska tuoteryhmälle ei voida määrittää mitään yhteistä kriteeriä, jonka perusteella niiden takuuasiat hoidettaisiin tietyllä tavalla. Suurin ongelma ei ole vastuualueiden määrän kasvaminen vaan niiden nimeäminen siten, että kaikki käyttäjät osaisivat valita oikean alueen tuotteen vastaanotossa.

Yksi ratkaisuvaihtoehto edellä kuvailtuun tilanteeseen on se, että luodaan yksi yleinen vastuualue tällaisia tapauksia varten ja vastuuhenkilöt muuttaisivat käsittelyiden vastualueet jälkikäteen. Tätä ei ole vielä alettu tehdä, koska tällöin vastuualueiden merkitys vähenisi lopulta olemattomaksi. Vastuualueen tarkoitus on toimia käsittelyn ohjaajana ja helpottaa tietojen kirjaustilannetta, mutta jos kaikki epäselvät tuotteet merkitään yhteiseen vastuualueeseen, asiakkaalle ei pystytäkään enää suoraan sanomaan, miten käsittely etenee. Tämä johtaa siihen, että käsittelyt viivästyvät ja vastuuhenkilöiden on itse huolehdittava, että yleisen alueen käsittelylista on mahdollisimman nopeasti tyhjä. Kun työmäärä ylittää vanhan käytännön, tulee järjestelmän tarpeellisuudesta ajan mittaan kyseenalainen.

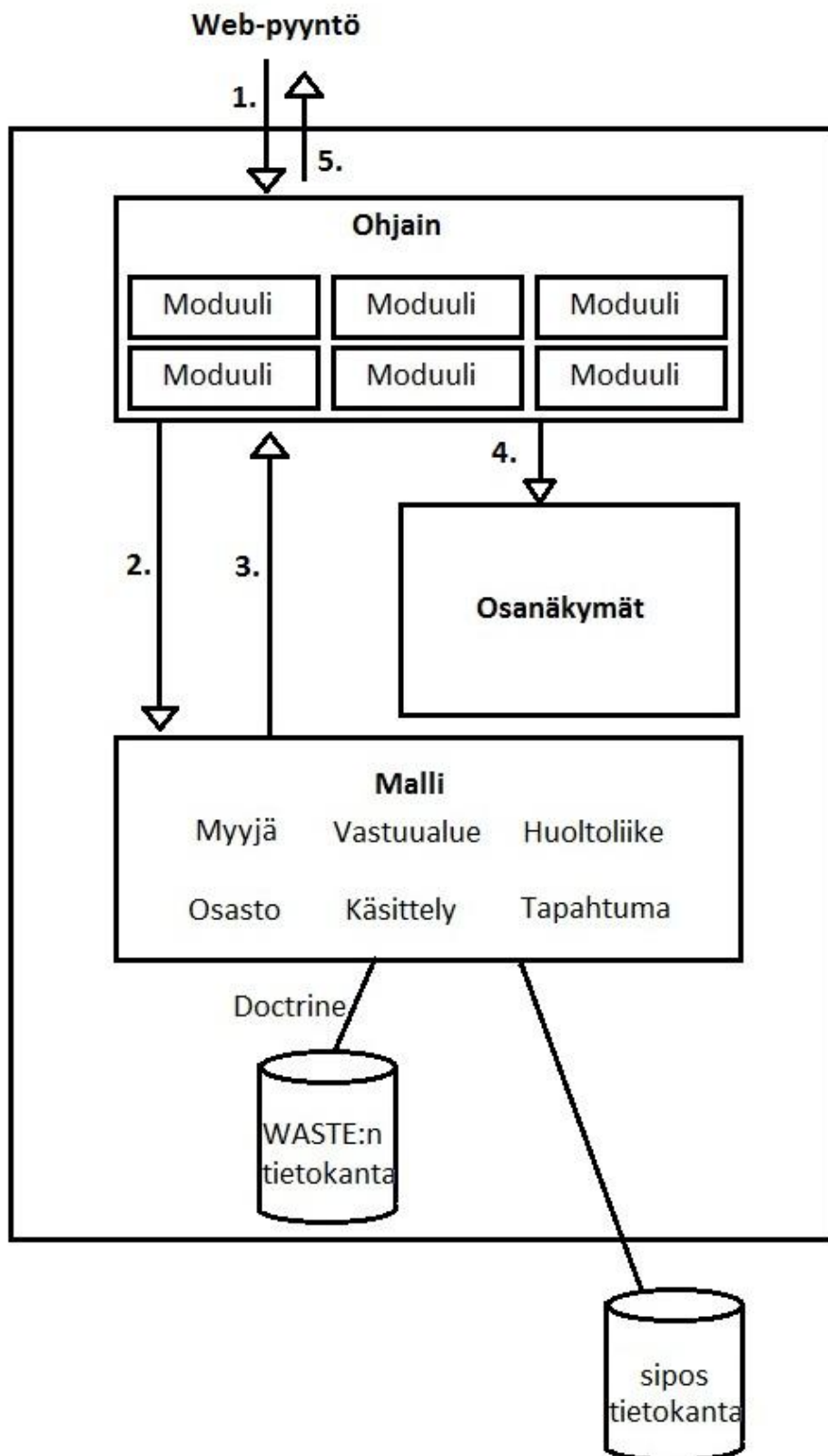
## LÄHTEET

Potencier, F. 2009. Practical symfony 1.3 & 1.4 for Doctrine. Luettu 10.9.2010  
[http://www.symfony-project.org/jobeeet/1\\_4/Doctrine/en/](http://www.symfony-project.org/jobeeet/1_4/Doctrine/en/)

Wage, J.H., Borschel, R., Blanco ,G. 2010. Doctrine ORM for PHP. Luettu 17.4.2011  
<http://www.doctrine-project.org/projects/orm/1.2/docs/manual/en>

Zaninotto, F. & Potencier, F. 2010. A Gentle Introduction to symfony 1.4. Luettu 10.9.2010  
[http://www.symfony-project.org/gentle-introduction/1\\_4/en/01-Introducing-Symfony](http://www.symfony-project.org/gentle-introduction/1_4/en/01-Introducing-Symfony)

|              |  |
|--------------|--|
| AJAX         | Asynchronous Javascript And XML, Javascript-tekniikka asynkroniseen tiedonsiirtoon   |
| CSRF         | Cross-Site Request Forgery, tietoturva-aukko, joka hyödyntää WWW-palvelimen luottamusta käyttäjän oikeellisuuteen                  |
| Doctrine     | symfonyn käyttämä ORM  |
| EAN          | International Article Number, eurooppalainen viivakoodistandardi   |
| LAMP         | Linux Apache MySQL PHP, ilmainen ja avoin web-sovellusalusta   |
| MIT-lisenssi | avoimen lähdekoodin lisenssi   |
| MVC          | model-view-controller, ohjelmoinnissa käytetty suunnittelumalli  |
| ORM          | Object-relational mapping, tekniikka, joka mahdollistaa relaatiotietokannan oliomaisen käsittelyn                                  |
| PHP          | web-ohjelmoinnissa paljon käytetty skriptikieli  |
| RMA          | Return merchandise authorization, tavarantoimittajan antama viitenumero tuotteen huoltoonlähetystä varten                          |
| rsync        | tiedostojen kopiointiin käytettävä työkalu   |
| SHA1         | kryptografinen tiivistefunktio   |
| SQL          | IBM:n kehittämä standardoitu tietokantojen kyselykieli   |
| XSS          | Cross-site scripting, WWW-sovelluksissa esiintyvä tietoturva-aukko, joka mahdollistaa haitallisen koodin syöttämisen verkkosivulle |
| YAML         | symfonyn asetustiedostoissa käytetty merkintäkieli   |



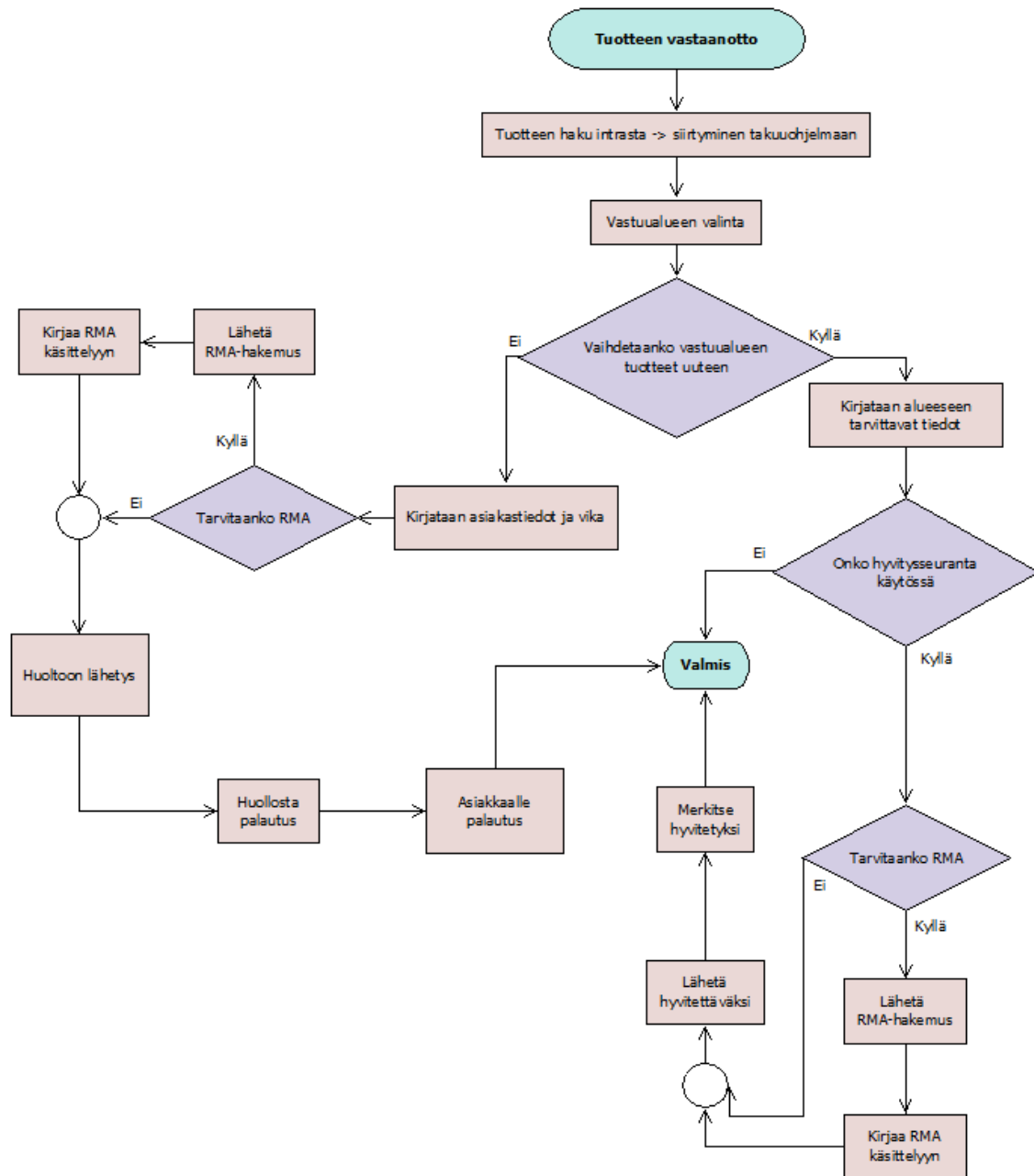
KUVIO 8: WASTE:n rakenne ja toiminta

TAULUKKO 1: Käyttäjätilin luomisessa testattavat toiminnot

| Testitoiminto   | Odotettu tulos   |
|---|--|
| Siirrytään järjestelmän pääsivulle  | Käsittelylistanäkymä   |
| Valitaan yläpalkista kohta Ylläpito   | Valittavana huoltoliikkeet, käyttäjät, osastot, vastuualueet                 |
| Siirrytään kohtaan käyttäjät  | Lista kaikista käyttäjistä   |
| Valitaan linkki "luo uusi tili"   | Tyhjä käyttäjätietolomake  |
| Valitaan tallenna syöttämättä mitään tietoja  | Pakollisten tietojen kenttien yläpuolelle tulostuu teksti "Pakollinen"       |
| Täytetään tiedot ja annetaan sähköpostiosoite virheellisessä muodossa (esim. ilman @-merkkiä) | Sähköpostikentän yläpuolelle tulostuu teksti "Virheellinen sähköpostiosoite" |
| Täytetään tiedot oikein ja valitaan tallenna  | Yhteenveto luodun käyttäjätilin tiedoista                                    |
| Palataan käyttäjälis-taan   | Listaan ilmestynyt luotu käyttäjä  |

## AKTIVITEETTIKAAVIO

## LIITE 4



KUVIO 9: Aktiviteettikaavio käsittelyn etenemisestä